

# Anonymous Post-Quantum Cryptocash

Huang Zhang<sup>1,2</sup>, Fangguo Zhang<sup>1,2</sup> \*, Haibo Tian<sup>1,2</sup>, and Man Ho Au<sup>3</sup>

<sup>1</sup> School of Data and Computer Science, Sun Yat-Sen University,  
Guangzhou 510006, China

<sup>2</sup> Guangdong Key Laboratory of Information Security,  
Guangzhou 510006, China

<sup>3</sup> Department of Computing, The Hong Kong Polytechnic University,  
Hong Kong, China

**Abstract.** In this paper, we construct an anonymous and decentralized cryptocash protocol which is secure in the quantum computation model. In order to achieve that, a linkable ring signature based on the ideal lattice is proposed. The size of a signature in our scheme is  $O(\log N)$ , where  $N$  is the number of participants in the ring. The framework of our cryptocash system follows that of CryptoNote with some modifications. By adopting the logarithmic size quantum resistant linkable ring signature scheme, our protocol is efficient and anonymous. We also introduce how to generate the verifying and signing key pairs of the linkable ring signature temporarily. With these techniques, both the sender and the receiver's privacy in transactions are protected even though they are published in the public ledger.

## 1 Introduction

Electronic currencies or cryptocash systems have been proposed for many years. But none of them are prevalent before the Bitcoin system appears. Bitcoin was first described by Satoshi Nakamoto in 2008 [23]. Its success is partially due to its properties of decentralization and anonymity. To prevent “double spending”, the system maintains the history of transactions among most nodes in a peer-to-peer network. A consensus mechanism called proof-of-work is used to maintain the history.

Later, researchers find that the public history of Bitcoin causes weaknesses which violate its original designing goals. The latest result states that Bitcoin only addresses the anonymity and unlinkability issues [3] partially. For example, multiple public keys of the same user can potentially be linked when a user gets changes back, in which case two or more of a single user's public keys will appear in the same transaction [25]. Recently, there are more discussions about the weak anonymity of Bitcoin [24, 27]. Although this weakness can be overcome by adopting “laundering services” and distributed methods, the solutions have to include a trusted third party which is a violation to the decentralization property.

---

\* Corresponding author, email: [isszhfg@mail.sysu.edu.cn](mailto:isszhfg@mail.sysu.edu.cn)

There are some creative works to design a strong anonymous cryptocash system. Miers *et al.* proposed “ZeroCoin” that allows users to spend their coins using anonymous proof of ownership instead of explicit public-key based digital signatures [22]. Sabherhagen presented two properties, namely, “untraceability” and “unlinkability”, which a fully anonymous cryptocash model must satisfy. They designed the “CryptoNote” system with these properties [28]. Monero is a system based on CryptoNote. In CryptoNote, to provide anonymity, there are two ways for all transactions on the network: (1) hiding the sender’s address using ring signatures, (2) hiding the receiver’s address using stealth addresses. Both sending and receiving addresses are public keys of a ring signature scheme. The ring signature can also be used in the ZeroCoin system [12].

The ring signature, introduced by Rivest *et al.* [26], permits a user to sign the message on behalf of a group. A verifier is convinced that the real signer is a member of a group, but cannot explicitly identify the real signer. Considering the anonymity of a cryptocash system, a ring signature is obviously more suitable than a standard signature. But there is a cost: the size of the signature and the computational complexity are inherently larger than that of a standard signature. A traditional ring signature scheme usually features a signature size of  $O(N)$ , where the ring has  $N$  participants. To construct a ring signature of  $O(\log N)$  or  $O(1)$  size was an open problem in this field. Recently, Groth and Kohlweiss proposed a DLP-based scheme with logarithmic size in the cardinality of the ring [12].

However, a cryptocash system which naively replace a standard signature with a ring signature suffered from the double spending attack. To fix this problem, it is necessary for the public to determine ring signatures generated by the same key pair at least. The traceable ring signature [9], which is modified as a “one-time signature” and adopted in CryptoNote and Monero etc<sup>4</sup>, provides the ability to trace the verifying and signing key pair which have been used to sign different messages. In general, a linkable ring signature [16], which is a variant of the linkable spontaneous anonymous group signature [15] and merely determines the double spending, is strong enough for cryptocash systems. Even though a signature of these schemes are of size  $O(N)$ , CryptoNote and Monero do provide better privacy than Bitcoin.

Most cryptocash systems are based on classic cryptographic schemes. The security of these schemes is based on hard mathematical problems, such as the factorization or discrete logarithm problems (DLP). However, researchers believe that a quantum computer is able to solve these problems efficiently and thus it can be used to break schemes based on these problems. One solution is to build schemes on mathematical problems that remain even hard for quantum computers. lattice-based problems are widely believed as suitable choices to build quantum resistant cryptographic schemes since Ajtai brought it into the region of cryptol-

---

<sup>4</sup> Recently, the organization of Monero posted the risk of double-spending with respect to all CryptoNote-based cryptocurrencies. What leads to this problem is the underlying elliptic curve group rather than the disability of the traceable ring signature itself.

ogy [2]. Some post-quantum signature schemes have been proposed recently [17, 8, 10]. Relying on these schemes, it is easy to obtain a post-quantum cryptocash system by replacing the ECDSA signature scheme in Bitcoin. However, the resulting cryptocash system is simply like Bitcoin in that their transactions are still linkable. Even though there are several lattice-based ring signatures [5, 7, 30, 31] including the one with logarithmic size [14], none of them has the linkable or traceable property which is vital to prevent double spending.

In this paper, we aim at designing an anonymous post quantum cryptocash (APQC) system. In order to achieve the goal, we propose a linkable ring signature (LRS) based on ideal lattices. The signature size of this scheme is  $O(\log N)$ , where  $N$  is the cardinality of the ring. The framework of our cryptocash protocol follows that of CryptoNote [28], and the lattice-based signature scheme is inspired by the work of Groth and Kohlweiss [12] with some modifications.

The paper is organized as follows: in Sect.2, we introduce notations and concepts used in our work; The model of the ring signature based cryptosystem is described in Sect.3; Section 4 involves the concrete construction of the lattice-based linkable ring signatures; we design the standard transaction of our cryptocash system in Sect.5; In Sect.6, we will give a brief conclusion and discuss our future works.

## 2 Preliminaries

### 2.1 Notations

We use  $\mathbb{Z}$ ,  $\mathbb{R}$ ,  $\mathbb{F}$  to denote the set of all integers, the set of all reals, and a field respectively. For any  $x \in \mathbb{R}$ ,  $\lceil x \rceil$  denotes the smallest integer that is not smaller than  $x$ . Column vectors are named by lower-case bold letters (e.g.,  $\mathbf{x}$ ) and matrices by upper-case bold letters (e.g.,  $\mathbf{X}$ ). For a vector  $\mathbf{x}$ ,  $\|\mathbf{x}\|_p$  represents its  $\ell_p$  norm, and  $p$  is omitted if  $p = \infty$ . The norm of a polynomial is defined similarly by regarding it as a vector. If  $\mathbf{x}$  is a vector of polynomial, then  $\|\mathbf{x}\| = \max_{x_i \in \mathbf{x}} \{x_i\}$ . For a matrix  $\mathbf{X}$ , define  $\|\mathbf{X}\|_p = \max_{\mathbf{y} \in \mathbf{X}} (\|\mathbf{y}\|_p)$ . If  $a \in R$  and  $\mathbf{X}$  is a matrix with entries in ring  $R$ ,  $a\mathbf{X}$  denotes the scalar multiplication.

Let  $x$  be any symbol,  $\{x_i\}_{i=1}^n$  denotes the set  $\{x_1, \dots, x_n\}$ .  $\mathbf{I}$  is the identity matrix whose dimension is known from the context. For an integer  $i$ ,  $i_j$  symbolizes the  $j^{\text{th}}$  bit of  $i$ .  $\delta_{i\ell}$  is Kronecker's delta, i.e.,  $\delta_{\ell\ell} = 1$  and  $\delta_{i\ell} = 0$  for  $i \neq \ell$ . For two strings  $x_1$  and  $x_2$ ,  $x_1\|x_2$  denotes the concatenation of them.

### 2.2 Lattices and Hard problems

A lattice  $\Lambda = \mathcal{L}(\mathbf{B})$  with dimension  $m$  and rank  $n$  is a subgroup of the linear space  $\mathbb{R}^m$ . Every element in  $\Lambda$  can be represented as an integral combination of its basis  $\mathbf{B} \in \mathbb{R}^{m \times n}$ . In our work, we will focus on a specific class of lattices, called ideal lattices, which can be described as ideals of certain polynomial rings.

**Definition 1** ([18]). *An ideal lattice is an integer lattice  $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$  such that  $\mathcal{L}(\mathbf{B}) = \{g \bmod f : g \in I\}$  for some monic polynomial  $f$  of degree  $n$  and ideal  $I \in \mathbb{Z}[x]/\langle f \rangle$ .*

The quotient ring  $\mathbb{Z}[x]/\langle f \rangle$  is isomorphic to the integer lattice  $\mathbb{Z}^n$ .

To extend the the hash function family in previous works [2, 6, 21], Micciancio defined the generalized knapsack function family.

**Definition 2 ([20]).** For any ring  $R$ , subset  $D \subset R$  and integer  $m \geq 1$ , the generalized knapsack function family  $\mathcal{H}(R, D, m) = \{f_a : D^m \rightarrow R\}_{a \in R^m}$  is defined by

$$f_a(x) = \sum_{i=1}^m x_i \cdot a_i,$$

for all  $a \in R^m$  and  $x \in D^m$ , where  $\sum_i x_i a_i$  is computed using the ring addition and multiplication operations.

In the process of proving the one-way property of their function, Micciancio showed that for a special case of the generalized knapsack function family, the distribution of  $f_a(x)$  is uniform and independent from  $a$ .

**Theorem 1 ([20]).** For any finite field  $\mathbb{F}$ , subset  $S \subset \mathbb{F}$ , and integers  $n, m$ , the hash function family  $\mathcal{H}(\mathbb{F}^n, S^n, m)$  is  $\epsilon$ -regular for

$$\epsilon = \frac{1}{2} \sqrt{(1 + |\mathbb{F}|/|S|^m)^n - 1}.$$

In particular, for any  $q = n^{O(1)}$ ,  $|S| = n^{\Omega(1)}$  and  $m = \omega(1)$ , the function ensemble  $\mathcal{H}(\mathbb{F}_q^n, S^n, m)$  is almost regular.

Here,  $\epsilon$ -regular means that the statistical distance between uniform distribution  $U((\mathbb{F}^n)^m, \mathbb{F}^n)$  and  $\{(a, f_a(x)) : a \in (\mathbb{F}^n)^m, x \in (S^n)^m\}$  is at most  $\epsilon$ . Note that  $(\mathbb{F}^n)^m$  is isomorphic to the quotient ring  $R = \mathbb{F}[x]/\langle f \rangle$  with a monic polynomial  $f \in \mathbb{F}[x]$ .  $S^n$  can be regarded as the subset of  $R$ . Subsequently, the above hash function family is a special case of the generalized knapsack function.

Sometimes, the one way property of a function is not sufficient for the design of a cryptographic protocol. Lyubashevsky and Micciancio proved that to find a collision in some instance of the Generalized Compact Knapsacks function is as hard as solving the worst-case problem in a certain lattice [19].

**Definition 3.** Let  $R = \mathbb{Z}[x]_q/\langle f \rangle$  be a ring for some integer  $q$ , where  $f \in \mathbb{Z}[x]$  is monic and irreducible polynomial of degree  $n$  and expansion factor  $EF(f, 3) \leq \epsilon$ . Let  $D = \{g \in R : \|g\| \leq \beta\}$  for some positive integer  $\beta$ .  $m \geq 1$  is a positive integer. Consider the instance of the Generalized Compact Knapsacks function family  $H$  which is a collection of functions  $h_a : D^m \rightarrow R$  indexed by  $a \in R^m$ . For  $b \in D^m$ ,  $h_a(b) = \sum_{i=1}^m b_i \cdot a_i$ . The collision problem  $Col_H$  is as follows: given a function  $h \in H$ , find  $b, c \in D^m$  such that  $h_a(b) = h_a(c)$ .

The expansion factor is a parameter proposed to quantify the quality of  $f$  which is the modulus in the ideal lattice [19]. The expansion factor of  $f$  is defined as

$$EF(f, k) = \max_{g \in \mathbb{Z}[x], \deg(g) \leq k(\deg(f)-1)} \|g\|_f / \|g\|_\infty$$

where  $\|g\|_f$  is short for  $\|g \bmod f\|_\infty$ . Moreover,  $EF(x^n + 1, k) \leq k$ .

**Theorem 2 ([19]).** *Let  $H$  be a Generalized Compact Knapsacks function family with  $m \geq \frac{\log q}{\log 2^\beta}$  and  $q > 2\varepsilon\beta mn^{1.5} \log n$ . Then, for  $\gamma = 8\varepsilon^2\beta mn \log^2 n$ , there is a polynomial time reduction from  $f\text{-SPP}_\gamma(I)$  for any ideal  $I \in R$  to  $\text{Col}_H(h)$  where  $h$  is chosen uniformly random from  $H$ . Furthermore, there is a straightforward reduction from  $I(f)\text{-SVP}_\gamma$  to  $f\text{-SPP}_\gamma$  (and also the other way around).*

The  $I(f)\text{-SVP}_\gamma$  with a polynomial approximating factor  $\gamma$  is widely believed to be intractable against even a quantum computer.

### 2.3 The Public-key Encryption on Ideal Lattices

The cryptosystem we described here was proposed by Stehlé *et al.* [29]. The ideal-lattice-based encryption scheme is formalized as a tuple of efficient procedures  $\mathcal{ES} = (\text{Setup}, \text{KGen}, \text{Enc}, \text{Dec})$ .

**Setup**( $1^n$ ):  $n$  is the security parameter. Fix  $f(x) = x^n + 1$  and  $q = \text{poly}(n)$  a prime satisfying  $q \equiv 3 \pmod{8}$ . Set  $\sigma = 1$ ,  $r = 1 + \log_3 q$ , and  $m = (\lceil \log q \rceil + 1)\sigma + r$ . Let  $R = \mathbb{Z}_q[x]/\langle f \rangle$ . All the parameters generated in this procedure are published as the global parameter  $pp$ .

**KGen**( $pp$ ): On input global parameter  $pp$ , it runs the trapdoor generation algorithm **Id-Trap** to get a one way function  $h_g : \mathbb{Z}_q^n \times \mathbb{Z}_q^{mn} \rightarrow \mathbb{Z}_q^{mn}$ . The first component of the domain of  $h_g$  can be viewed as a subset of  $\mathbb{Z}_2^{\ell_I}$  for  $\ell_I = O(n \log q) = \tilde{O}(n)$ . Generate  $r \in \mathbb{Z}_2^{\ell_I + \ell_\mu}$  uniformly and define the Toeplitz matrix  $M_{GL} \in \mathbb{Z}_2^{\ell_\mu \times \ell_i}$  whose  $i^{\text{th}}$  row is  $(r_i, \dots, r_{\ell_I + i - 1})$ . It outputs public key  $epk = (g, r)$  and the secret key is  $esk = S$ .

**Enc**( $pp, epk, \mu$ ): Given  $\ell_\mu$  bit message  $\mu$  with  $\ell_\mu = n / \log n$  and public key  $epk = (g, r)$ , sample  $(s, e)$  with  $s \in \mathbb{Z}_q^n$  uniform and  $e$  sampled from  $\bar{\Psi}_{\alpha q}$ , where  $\bar{\Psi}_{\alpha q}$  is the reduction modulo  $q$  of the standard Gaussian distribution with parameter  $\alpha q$ . It then evaluates  $C_1 = h_g(s, e)$  and computes  $C_2 = \mu \oplus (M_{GL} \cdot s)$ , where the product is computed over  $\mathbb{Z}_2$ , and  $s$  is viewed as a string over  $\mathbb{Z}_2^{\ell_I}$ . Return ciphertext  $C = (C_1, C_2)$ .

**Dec**( $pp, esk, c$ ): Given ciphertext  $C = (C_1, C_2)$  and secret key  $esk = (S, r)$ , invert  $C_1$  to compute  $(s, e)$  such that  $h_g(s, e) = C_1$ , and return message  $\mu = C_2 \oplus (M_{GL} \cdot s)$ .

To see the details of the trapdoor generation algorithm **Id-Trap** and the one-way trapdoor function family  $\{h_g : \mathbb{Z}_q^n \times \mathbb{Z}_q^{mn} \rightarrow \mathbb{Z}_q^{mn}\}_{g \in (\mathbb{Z}_q[x]/f)^m}$ , we refer to the literature [29] in which Stehlé *et al.* also proved that the above encryption scheme is IND-CPA secure if the Ideal-LWE $_{m,q;\Psi_{\alpha q}}^f$  problem is hard.

**Theorem 3 ([29]).** *Any IND-CPA attacker against  $\mathcal{ES}$  with run-time  $T$  and success probability  $1/2 + \varepsilon$  provides an algorithm for Ideal-LWE $_{m,q;\Psi_{\alpha q}}^f$  with runtime  $O(2^{3\ell_\mu} n^3 \varepsilon^{-3} \cdot T)$  and success probability  $\Omega(2^{-\ell_\mu n^{-1} \cdot \varepsilon})$ .*

The notion of key privacy (also known as key indistinguishability and ciphertext anonymity) is formally defined by Bellare *et al.* [4]. It requires that the receiver of a ciphertext is anonymous from the point of view of the adversary. Due

to the conclusion of observation 1 proposed by Halevi [13], the aforementioned encryption scheme  $\mathcal{ES}$  is key privacy, since its parameters satisfy the requirement of observation 1.

### 3 Anonymous Cryptographic Currency Model Based on Linkable Ring Signatures

Cryptocash system based on linkable ring signatures emerged after researchers found that Bitcoin was not fully anonymous and untraceable. CryptoNote and Monero are two typical instances. We describe here the properties of an anonymous cryptocash system and state the techniques [28] to construct such a system.

In a cryptocash system, there are three parties: a sender, who owns a coin and decides to spend it, a receiver, who is the destination that a coin is delivered to, and a public ledger where all transactions are recorded. An anonymous cryptocash system should satisfy the following properties:

- Untraceability: If  $Tx$  is a transaction from sender  $A$  to receiver  $B$ , and  $Tx$  has been recorded in the public ledger, no one can determine the sender with probability significantly larger than  $1/N$  by accessing the transcript of  $Tx$ , where  $N$  is the number of possible senders in a related input of the  $Tx$ . Moreover, even receiver  $B$  cannot prove that  $A$  is the true sender of  $Tx$ .
- Unlinkability: If  $Tx_1$  is a transaction from sender  $A$  to receiver  $C$ ,  $Tx_2$  is another transaction from sender  $B$  to receiver  $C$ , and  $Tx_1, Tx_2$  have been recorded in the public ledger, then for any subsequent transactions in the public ledger, no one can use them to link the outputs of the two transactions to a single user, even for senders  $A$  and  $B$ .
- Detecting Double Spending: If  $Tx_1$  is a transaction which describes that coin  $c$  has been sent from sender  $A$  to receiver  $B$ , and  $Tx_1$  has been recorded in the public ledger, every user of the system could detect another transaction  $Tx_2$  that describes the same coin  $c$ . Furthermore,  $Tx_2$  will never be accepted and recorded in the public ledger.

To design a cryptocash protocol which provides all the above properties, the CryptoNote and Monero suggested to adopt the modification of the traceable ring signature [9], which generates a one time signature on behalf of a temporal group. Since it is a one time signature with an explicit identification tag about the signing key, it could prevent a coin being double-spent. Besides, since it is a ring signature where the identity of the real signer is hidden within a set of possible signers, it guarantees untraceability. In addition, ring signature supports unlinkability since the inputs in a transaction may be brought from outputs of transactions belonging to other users.

To employ a linkable ring signature in a cryptocash system, the receiver should produce a one-time key pair for each transaction. A sender could obtain the public key of the receiver for the transaction and build a transaction with an output script containing that key's information. The drawback of this trivial method is that a receiver has to maintain a lot of one-time keys. Furthermore, a sender has to

contact each receiver for their fresh one-time public key when the sender builds a transaction. Alternatively, CryptoNote suggests another method which enables a receiver to store only a single key pair. A sender could produce a random value to generate a one-time public key for the receiver based on this single public key. The one-time public key is referred to as the destination address. This is a convenient design at the cost of a slightly weakened unlinkability. Specifically, if a user has a single key, a sender could always identify a receiver from the sender's transaction by its random value of the transaction. If two senders collude, and they have sent coins to the same receiver, they could identify the same receiver while the trivial method avoids this. And if a later transaction includes the two senders' outputs at the same time, with a higher probability, the later transaction is made by the receiver. Note that a receiver could still produce another key pair at will as in the Bitcoin system to avoid the small problem.

Finally, let us observe a standard transaction in a linkable ring signature based cryptocash system. In such a system, the value of a coin is bound with a destination address. Suppose  $A$  and  $B$  are two users in the system.  $B$  has a single key pair  $(pk_B, sk_B)$ .  $A$  has the private key  $sk_1$  of a destination address  $vk_1$ , which represents a coin, say  $c$ , which has been sent to  $A$  previously. If  $A$  decides to send  $c$  to  $B$ , he generates a destination address  $vk_2$  and an auxiliary input  $aux$  for  $B$ ; he then chooses a number of transactions from the public ledger such that the delivered value of coin is equivalent to  $c$ ; he extracts the destination addresses of those transactions and assembles them with  $vk_1$  to form a ring  $L$ ; he runs a ring signature algorithm to sign transaction  $Tx$ , which involves information about  $(c, aux, vk_2, L)$ , with signing key  $sk_1$  and broadcasts the transaction; If the signature generated by  $sk_1$  is not linkable to any transaction on the ledger, the public ledger will accept this transaction and record it;  $B$  uses its private key  $sk_B$  to check every passing transaction to determine if transaction  $Tx$  is for  $B$  and recovers the signing key  $sk_2$  corresponding to  $vk_2$ . With  $sk_2$ , user  $B$  can spend  $c$  by signing another transaction. However, even  $A$  does not know when and where  $B$  spends it due to the functionality of the linkable ring signature.

It is obvious that linkable ring signature is vital for an anonymous cryptocash system. We next detail the lattice-based version of a linkable ring signature.

## 4 Linkable Ring Signature Based on Ideal-Lattices

The strong similarity (see Lyubashevsky's signature [17] and the Schnorr signature) in the construction between a lattice-based signature and DLP based one implies that latter can help us design the lattice-based counterparts of DLP based schemes, e.g., using the work in [16] or [17], it can easily obtain a linkable ring signature based on lattices with signature size of  $O(N)$ , where  $N$  is the number of participants of the ring. However, such a construction is not efficient enough for a practical cryptocash protocol. In this section, we aim at presenting a linkable ring signature of size  $O(\log N)$  using the idea in [12].

#### 4.1 A Brief Recall on Previous Generic Construction

In [12], Groth and Kohlweiss proposed an efficient Sigma-protocol, which can actually be used as an ad-hoc group identification scheme. Their ring signature scheme is a direct transformation of the identification scheme with the Fiat-Shamir heuristic. As the transmission of the identification scheme involves only logarithmic number of commitments, the result ring signature scheme is of logarithmic size.

Their work started from homomorphic commitments scheme such as Pedersen commitments ( $\text{com}(m; r) = g^m h^r$ ). The first step is to design a Sigma-protocol  $\Sigma_1$  to prove in zero-knowledge that the homomorphic commitment is opened to 0 or 1. Once the subroutine  $\Sigma_1$  is established, to design an ad-hoc group identification scheme is to construct a Sigma-protocol  $\Sigma_2$  to show in zero-knowledge that one of the  $N$  foregoing commitments containing 0, where such  $N$  commitments, say  $\text{com}(0, r_i)$  for  $i \in \{0, \dots, N-1\}$ , are the public keys of  $N$  users and  $r_i$ 's are their secret keys respectively. Assume that the actual user to be identified is labeled by  $\ell$  for  $\ell \in \{0, \dots, N-1\}$ .  $\Sigma_2$  first implements  $\log N$  commitments to hide the number  $\ell$  bit by bit and prove in zero-knowledge that those commitments are opened to 0 or 1 by running  $\Sigma_1$  protocol  $\log N$  times. Then  $\Sigma_2$  proves in knowledge, that the  $\ell^{\text{th}}$  user can open  $\text{com}(0, r_\ell)$  to 0, with the help of the intermediate parameters used in the foregoing  $\Sigma_1$ 's. By replacing the challenge message with the hash value of all initial messages in  $\Sigma_2$ , we obtain a non-interactive zero-knowledge proof system which can be regarded as a ring signature. For the details of the generic construction of  $\Sigma_2$ , we refer the reader to the literature [12].

It is worth mentioning that the underlying homomorphic commitment is the corner stone of both the construction and security proof of the foregoing ring signature. As a counterpart of their work, our scheme also contains a lattice-based commitment (e.g.,  $\text{com}(\mathbf{S}; \mathbf{X}) = \mathbf{HS} + \mathbf{GX}$ ). The details of our commitment scheme is left to Sect.4.3.

#### 4.2 Our Construction

To construct an  $O(\log N)$  ring signature, Groth and Kohlweiss proposed a technique to compute the coefficients of a polynomial in the indeterminate  $x$  over the finite field  $\mathbb{Z}_q$  in advance, where  $x$  is a hash value computed later [12]. We extend their method to handle the polynomial with coefficient in a ring of square matrices. The major difference is that the multiplication of the matrix ring is not commutative. This is the reason why we restrict  $x$  in our scheme to be a  $1 \times 1$  matrix. Since the scalar multiplication is commutative, we have the following result.

Given matrices  $\mathbf{B}_j$ , set  $\mathbf{W}_j = \ell_j x \mathbf{I} + \mathbf{B}_j$ , here  $\ell_j \in \{0, 1\}$ ,  $\mathbf{I}$  is the identity matrix. Let  $\mathbf{W}_{j,1} = \mathbf{W}_j = \ell_j x \mathbf{I} + \mathbf{B}_j = \delta_{1\ell_j} x \mathbf{I} + \mathbf{B}_j$  and  $\mathbf{W}_{j,0} = x \mathbf{I} - \mathbf{W}_j = (1 - \ell_j) x \mathbf{I} - \mathbf{B}_j = \delta_{0\ell_j} x \mathbf{I} - \mathbf{B}_j$ . Then for each  $i$ , the product  $\prod_{j=1}^n \mathbf{W}_{j,i_j}$  is a polynomial of the form

$$P_i(x) = \prod_{j=1}^n (\delta_{i_j \ell_j} x \mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{P}_{i,k} x^k = \delta_{i\ell} x^n \mathbf{I} + \sum_{k=0}^{M-1} \mathbf{P}_{i,k} x^k.$$

Hence,  $\mathbf{P}_{i,k}$  is the coefficient of the  $k^{\text{th}}$  degree term in the polynomial, and can be efficiently computed when  $\{\mathbf{B}_j\}_{j=1}^M$  and  $\ell$  are given.

The linkable ring signature scheme consists of a tuple of efficient procedures  $\mathcal{LRS} = (\text{Setup}, \text{KGen}, \text{Sign}, \text{Vfy}, \text{Link})$ . The details of them are shown as follows:

**Setup**( $1^n$ ): Let  $N$  be the maximum size of the ring and  $M = \log N$ ,  $\beta$  be the parameter of the Generalized Compact Knapsack problem,  $\mathcal{H} : \{0, 1\}^* \rightarrow \{\mathbf{v}, \mathbf{v} \in \{-1, 0, 1\}^n, \|\mathbf{v}\|_1 \leq p\}$  be a random oracle introduced in [17] such that  $2^p \cdot \binom{n}{p} \geq 2^{100}$  (which is typically set to be  $p = n^{\frac{1}{2}}$ ). On input security parameter  $n$ , set  $f = x^n + 1$ ,  $\varepsilon = 3$ ,  $t = O(n \log n)$ ,  $\beta \geq \frac{t(1-p^{M+1})}{(1-p)}$ ,  $m \geq 3$ . Pick a prime  $q$  such that  $(2\beta)^m > q > 2\varepsilon\beta mn^{1.5} \log n$ . All operations in this system are done in  $R = \mathbb{Z}_q[x]/\langle f \rangle$ . Let  $D = \{g \in R : \|g\| \leq t\}$ . Relying on those parameters, this procedure samples matrices  $\mathbf{G}, \mathbf{H} \in R^{1 \times m}$  uniformly at random. Finally it outputs  $pp = (n, m, \mathbf{G}, \mathbf{H}, \mathcal{H}, q, t, N)$  as the global parameters.

**KGen**( $pp$ ): For the  $i^{\text{th}}$  user, this procedure randomly chooses  $\mathbf{X}_i \leftarrow D^{m \times m}$  and computes  $\mathbf{Y}_i = \mathbf{G}\mathbf{X}_i$ . The  $i^{\text{th}}$  user's verifying key is  $vk_i = \mathbf{Y}_i$  and the signing key is  $sk_i = \mathbf{X}_i$ .

**Sign**( $pp, sk_\ell, \mu, L$ ): Without loss of generality, let  $L = (\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_{N-1})$  be the ensemble of a ring with the largest size,  $M = \lceil \log N \rceil$ . On input a message  $\mu$ , the  $\ell^{\text{th}}$  user's signature on behalf of  $L$  is generated as follows

- Compute  $\mathbf{R}_\ell = \mathbf{H}\mathbf{X}_\ell$ .
- For  $j$  from 1 to  $M$ ,
  - sample  $\mathbf{K}_j, \mathbf{C}_j, \mathbf{D}_j, \mathbf{E}_k \leftarrow D^{m \times m}$ ,
  - if  $\ell_j = 0$ , randomly pick  $\mathbf{B}_j \leftarrow D^{m \times m}$ ,
  - else if  $\ell_j = 1$ , draw  $\mathbf{B}_j \in \{g \in R : \|g\| \leq t - 1\}$  uniformly at random,
  - compute  $\mathbf{V}_{\ell_j} = \mathbf{H}(\ell_j \mathbf{I}) + \mathbf{G}\mathbf{K}_j$ , and  $\mathbf{V}_{a_j} = \mathbf{H}\mathbf{B}_j + \mathbf{G}\mathbf{C}_j$ ,
  - compute  $\mathbf{V}_{b_j} = \mathbf{H}(\ell_j \mathbf{B}_j) + \mathbf{G}\mathbf{D}_j$ ,
  - compute  $\mathbf{V}_{d_k} = (\sum_{i=0}^{N-1} \mathbf{Y}_i \mathbf{P}_{i,k}) + \mathbf{G}\mathbf{E}_k$ , where  $k = j - 1$ ,
  - compute  $\mathbf{V}'_{d_k} = \mathbf{H}\mathbf{E}_k$ , where  $k = j - 1$ .
- Let  $S_1 = \{\mathbf{V}_{\ell_j}, \mathbf{V}_{a_j}, \mathbf{V}_{b_j}, \mathbf{V}_{d_{j-1}}, \mathbf{V}'_{d_{j-1}}\}_{j=1}^M$  and compute  $x = \mathcal{H}(pp, \mu, L, S_1, \mathbf{R}_\ell)$ .
- For  $j$  from 1 to  $M$ , compute
  1.  $\mathbf{W}_j = \ell_j x \mathbf{I} + \mathbf{B}_j$ ,
  2.  $\mathbf{Z}_{a_j} = \mathbf{K}_j(x \mathbf{I}) + \mathbf{C}_j$ ,
  3.  $\mathbf{Z}_{b_j} = \mathbf{K}_j(x \mathbf{I} - \mathbf{W}_j) + \mathbf{D}_j$ ,
  4.  $\mathbf{Z}_d = \mathbf{X}_\ell(x^M \mathbf{I}) - \sum_{k=0}^{M-1} \mathbf{E}_k x^k$ .
- Let  $S_2 = \{\mathbf{W}_j, \mathbf{Z}_{a_j}, \mathbf{Z}_{b_j}\}_{j=1}^M$ . Publish  $\sigma = \{S_1, S_2, \mathbf{Z}_d, \mathbf{R}_\ell, L\}$  as the signature of the  $\ell^{\text{th}}$  user.

**Vfy**( $pp, \mu, \sigma$ ):

1. Compute hash value  $x = \mathcal{H}(pp, \mu, L, S_1, \mathbf{R}_\ell)$ .
2. For  $j$  from 1 to  $M$ , consider the following inequalities

- $\|\mathbf{W}_j\| \leq t$ ,
  - $\|\mathbf{Z}_{a_j}\| \leq (p+1)t$ ,
  - $\|\mathbf{Z}_{b_j}\| \leq tp + t^2nm + t$ ,
  - $\|\mathbf{Z}_d\| \leq t \frac{p^{M+1}-1}{p-1}$ . If any of them does not hold, output 0 and abort.
3. For  $j$  from 1 to  $M$ , consider following equations
- $\mathbf{V}_{\ell_j}(\mathbf{x}\mathbf{I}) + \mathbf{V}_{a_j} = \mathbf{H}\mathbf{W}_j + \mathbf{G}\mathbf{Z}_{a_j}$ ,
  - $\mathbf{V}_{\ell_j}(\mathbf{x}\mathbf{I} - \mathbf{W}_j) + \mathbf{V}_{b_j} = \mathbf{G}\mathbf{Z}_{b_j}$ .
- If any of the aforementioned equations does not hold, output 0 and abort.
4. If the equation  $\mathbf{R}_\ell(x^M\mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{V}'_{d_k}(-x^k) = \mathbf{H}\mathbf{Z}_d$  does not hold, output 0 and abort.
5. Inspect whether

$$\sum_{i=0}^{N-1} (\mathbf{Y}_i \prod_{j=1}^M \mathbf{W}_{j,i_j}) + \sum_{k=0}^{M-1} \mathbf{V}_{d_k}(-x^k) = \mathbf{G}\mathbf{Z}_d$$

holds. If not, output 0, otherwise output 1 (accept).

**Link**( $pp, \sigma_1, \sigma_2$ ): For two signatures  $\sigma_1 = (\dots, \mathbf{R}_1, L_1)$  and  $\sigma_2 = (\dots, \mathbf{R}_2, L_2)$ , if  $\mathbf{R}_1 = \mathbf{R}_2$ , return 1 (linked) for concluding that they are generated by the same signer; otherwise, return 0 (unlinked).

**Correctness:** To see that the signature generated by **Sign** procedure always passes the **Vfy** procedure, we first observe the four equations in the **Vfy** procedure. The equations in step 3 are to prove in zero-knowledge that the signer is the  $\ell^{th}$  user. The correctness of those equations is shown directly through a simple deduction. The equation in step 4 is to prove that the parameter for linking is correct. For a valid signature, it hold since

$$\mathbf{R}_\ell(x^M\mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{V}'_{d_k}(-x^k) = \mathbf{H}\mathbf{X}_\ell(x^M\mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{H}\mathbf{E}_k(-x^k) = \mathbf{H}(\mathbf{X}_\ell(x^M\mathbf{I}) - \sum_{k=0}^{M-1} \mathbf{E}_k x^k) = \mathbf{H}\mathbf{Z}_d$$

The equation in step 5 is to prove in zero-knowledge that the anonymous signer holds the secret key of the  $\ell^{th}$  user. To see the correctness of the last equation, note that  $\prod_{j=1}^M \mathbf{W}_{j,i_j}$  is a polynomial in the indeterminate  $x$ . Subsequently,  $\mathbf{Y}_\ell \prod_{j=1}^M \mathbf{W}_{j,i_j}$  yields a polynomial of degree  $n$ , while the other  $\mathbf{Y}_i \prod_{j=1}^M \mathbf{W}_{j,i_j}$ ,  $i \neq \ell$  leads to polynomials of degree  $n-1$ . Formally, we have

$$\begin{aligned} & \sum_{i=0}^{N-1} (\mathbf{Y}_i \prod_{j=1}^M \mathbf{W}_{j,i_j}) + \sum_{k=0}^{M-1} \mathbf{V}_{d_k}(-x^k) \\ &= \sum_{i=0}^{N-1} \mathbf{Y}_i (\delta_{i\ell} x^M \mathbf{I} + \sum_{k=0}^{M-1} \mathbf{P}_{i,k} x^k) + \sum_{k=0}^{M-1} ((\sum_{i=0}^{N-1} \mathbf{Y}_i \mathbf{P}_{i,k}) + \mathbf{G}\mathbf{E}_k)(-x^k) \\ &= \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} (\mathbf{Y}_i \mathbf{P}_{i,k} x^k - \mathbf{Y}_i \mathbf{P}_{i,k} x^k) + \mathbf{Y}_\ell (\delta_{\ell\ell} x^M \mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{G}\mathbf{E}_k(-x^k) \\ &= \mathbf{G}(\mathbf{X}_\ell(x^M\mathbf{I}) - \sum_{k=0}^{M-1} \mathbf{E}_k x^k) \\ &= \mathbf{G}\mathbf{Z}_d \end{aligned}$$

It remains to show that  $\{\mathbf{W}_j, \mathbf{Z}_{a_j}, \mathbf{Z}_{b_j}\}_{j=1}^M$ , and  $\mathbf{Z}_d$  are short enough to pass step 2 of the  $\mathbf{Vfy}$  procedure.

We note that for polynomials  $a, b \in R$ , the norm of their product is bounded by  $\|a\| \cdot \|b\| \cdot n$ . For  $a \in R$  and  $b \in \{v : v \in \{-1, 0, 1\}^n, \|v\|_1 \leq p\}$  the norm of  $a \cdot b$  is not larger than  $\|a\| \cdot \|b\| \cdot p$ . With the help of above two facts and the triangle inequality, the correctness of the norm about those matrices can be validated easily. For example,  $\|\mathbf{Z}_{b_j}\| = \|\mathbf{K}_j(x\mathbf{I} - \mathbf{W}_j) + \mathbf{D}_j\| \leq \|\mathbf{K}_j x \mathbf{I}\| + \|\mathbf{K}_j(-\mathbf{W}_j)\| + \|\mathbf{D}_j\| \leq tp + t^2 nm + t$  and  $\|\mathbf{Z}_d\| \leq \|\mathbf{X}_\ell(x^M \mathbf{I})\| + \|\sum_{k=0}^{M-1} \mathbf{E}_k x^k\| \leq tp^M + \|\mathbf{E}_0 x^0\| + \|\mathbf{E}_1 x^1\| + \dots + \|\mathbf{E}_{M-1} x^{M-1}\| \leq tp^M + t + tp + \dots + tp^{M-1} = t \frac{p^{M+1} - 1}{p - 1}$ .

Even though the forgoing linkable ring signature is designed on the hard problem of ideal lattice, a classic edition of this signature can be built by instead using any cyclic group as long as its underlying DLP is hard. We will propose a linkable ring signature based on the ECDLP, and discuss how to implement this signature with ECC, in Appendix A.

### 4.3 Security Proof

Groth and Kohlweiss have proved that the generic construction of their ring signature is secure in the random oracle model, if its underlying commitment scheme is perfectly hiding and computationally binding [12]. Since our linkable ring signature is designed over the framework of their generic construction, to prove the security of our scheme it is sufficient to prove the binding and hiding properties of the commitment adopted in our signature scheme.

**Theorem 4 (Anonymity and Unforgeability).** *The generic construction of the ring signature scheme in [12] is perfect anonymity if the underlying commitment scheme is perfectly hiding. It is unforgeable in the random oracle model if the commitment scheme is perfectly hiding and computationally binding.*

A non-interactive commitment scheme allows a sender to construct a commitment to a value. The sender may later open the commitment and reveal the value so that the receiver can verify the opening and check if it was the value which was committed at the beginning. A commitment scheme is said to be hiding, only if it reveals nothing about the committed value. The binding property ensures that a sender cannot open the commitment to two different values.

We now proceed to introduce the details of the underlying commitment in our  $\mathcal{LRS}$ . The non-interactive commitment scheme adopted in our  $\mathcal{LRS}$  consists of a pair of PPT algorithms  $\mathcal{CMT} = (\mathbf{Gen}, \mathbf{Com})$ .

**Gen( $1^n$ ):** The setup algorithm is tightly associated with that of  $\mathcal{LRS}$ . It runs  $\mathcal{LRS.Setup}(1^n)$  to get the global parameters  $\mathcal{LRS.pp}$  of the signature scheme and picks  $pp = (n, m, \mathbf{G}, \mathbf{H}, q, t)$  out of  $\mathcal{LRS.pp}$  to be global parameters of the commitment scheme. The value to be committed and the randomness to be chosen are uniformly sampled from  $D^{m \times m}$ .

**Com( $pp, \mathbf{S}$ ):** If a sender wants to construct a commitment  $\mathbf{C}$  to matrix  $\mathbf{S} \in D^{m \times m}$ , it uniformly samples  $\mathbf{X} \leftarrow D^{m \times m}$  and computes  $\mathbf{C} = \mathbf{HS} + \mathbf{GX}$ . The

commitment  $\mathbf{C}$  can later be opened by unveiling the short  $\mathbf{S}$  and  $\mathbf{X}$  where  $\|\mathbf{S}\|, \|\mathbf{X}\| \leq t$ .

The correctness of the forgoing commitment scheme  $\mathcal{CMT}$  is obvious. It remains to prove that  $\mathcal{CMT}$  is hiding and binding.

**Theorem 5 (Binding and Hiding).** *For any committed matrix  $\mathbf{S} \leftarrow D^{m \times m}$  and any random matrix  $\mathbf{X} \leftarrow D^{m \times m}$ , the commitment  $\mathbf{C} = \mathbf{HS} + \mathbf{GX}$  reveals nothing about  $\mathbf{S}$ . Moreover, the sender can't open the commitment  $\mathbf{C}$  to  $\mathbf{S}' \neq \mathbf{S}$ , if the collision problem  $Col_H$  defined in Definition 3 is hard.*

*Proof.* By recalling the definition of generalized knapsack function family, it is easy to see that  $\mathbf{G}$  fixes a function of the function family  $f_{\mathbf{G}} : D^m \rightarrow R$ , where  $D = \{g \in R : \|g\| \leq t\}$ . Let  $\mathbf{X}_i$  symbolizes the  $i^{th}$  column of the matrix  $\mathbf{X}$ . Since the coefficients of a polynomial in  $R$  belong to  $\mathbb{Z}_q$  which is a finite field, and since we set the global parameters  $q \geq 2\varepsilon\beta mn^{1.5} \log n$ ,  $t = O(n \log n)$ ,  $m \geq 3$ , the distribution of  $f_{\mathbf{G}}(\mathbf{X}_i) = \mathbf{GX}_i$  is almost uniform over  $\mathbb{Z}_q^n$  (which is essentially  $R$ ), due to Theorem 1. Subsequently,  $f_{\mathbf{G}}(\mathbf{X}) = \mathbf{GX}$  is almost uniform over  $R^m$ . Note that, this fact is also suitable for the product  $\mathbf{HS}$ . As a result,  $\mathbf{C} = \mathbf{HS} + \mathbf{GX}$  is uniform over  $R^m$  and hence  $\mathbf{C}$  reveals nothing about the committed value  $\mathbf{S}$ .

We proceed to prove the binding property. Because  $(2\beta)^m \geq q$ , we have  $m > \frac{\log q}{\log 2\beta}$ . Depending on Theorem 2, to find a collision in the generalized knapsack function  $f_{\mathbf{G}}(\mathbf{X})$  is as hard as to solve the  $I(f)$ -SVP $_{\gamma}$  problem, so is the function  $f_{\mathbf{H}}(\mathbf{S})$ . Here,  $\gamma = 8\varepsilon^2\beta mn \log^2 n$  is a polynomial in security parameter  $n$ . It is conjectured that approximating SVP to within a polynomial factor is a hard problem, although it is not NP-hard [1, 11].

Suppose, for sake of contradiction, that the sender can open the commitment  $\mathbf{C}$  to  $\mathbf{S}, \mathbf{S}' \in D^{m \times m}$  such that  $\mathbf{S} \neq \mathbf{S}'$ ,  $\|\mathbf{S}\| \leq t$ ,  $\|\mathbf{S}'\| \leq t$  and  $\mathbf{HS} + \mathbf{GX} = \mathbf{HS}' + \mathbf{GX}'$ .

We consider the two possible cases.

**Case 1:** If  $\mathbf{HS} = \mathbf{HS}'$ , then  $\mathbf{H}(\mathbf{S} - \mathbf{S}') = \mathbf{0}$ . Since  $\|\mathbf{S} - \mathbf{S}'\| \leq \|\mathbf{S}\| + \|\mathbf{S}'\| \leq 2t < \beta$ .  $\mathbf{S}$  and  $\mathbf{S}'$  are a pair of collisions of the function  $f_{\mathbf{H}}(\mathbf{S})$  which yield a contradiction to the hardness assumption introduced in Theorem 2.

**Case 2:** If  $\mathbf{HS} \neq \mathbf{HS}'$ , then  $\mathbf{X} \neq \mathbf{X}'$  and we have  $\mathbf{H}(\mathbf{S} - \mathbf{S}') = \mathbf{G}(\mathbf{X}' - \mathbf{X})$ . Similar to the forgoing discussions,  $\mathbf{S}$  and  $\mathbf{S}'$  yields a contradiction to the fact that  $f_{\mathbf{H}}(\mathbf{S})$  is collision resistant.

Depending on the discussions made in the **Case 1** and the **Case 2**, we have shown that the sender cannot open the commitment  $\mathbf{C}$  to different values.  $\square$

Since  $\mathcal{CMT}$  is binding and hiding, the anonymity and unforgeability of the linkable ring signature  $\mathcal{LRS}$  can be proved through the framework described in [12].

The next is to prove that our linkable ring signature is linkable.

**Theorem 6 (Linkability).** *Our linkable ring signature  $\mathcal{LRS}$  is linkable. Formally, given a set of signing keys  $S = \{\mathbf{X}_0, \dots, \mathbf{X}_{N-1}\}$ , it is impossible to produce  $N + 1$  signatures  $\sigma_0, \dots, \sigma_N$ , such that any two of them can pass the **Link** procedure.*

*Proof.* Suppose, for sake of contradiction, an adversary can produce  $N + 1$  valid signatures  $\sigma_i = \{S_1^{(i)}, S_2^{(i)}, Z_d^{(i)}, \mathbf{R}_i, L_i\}$  such that  $\mathbf{R}_i$ 's are pairwise distinct, for  $i \in \{0, 1, \dots, N\}$ . Since  $|S| = N$ , there is at least one  $\mathbf{R}_i$  which does not belong to the set  $\{\mathbf{H}\mathbf{X}_j : \mathbf{X}_j \in S\}$ . Without loss of generality, consider this event happened in  $\sigma_\pi$ , for  $\pi \in \{0, \dots, N\}$ . As  $\sigma_\pi$  is a valid signature, we have  $\mathbf{R}_\pi(x^M \mathbf{I}) + \sum_{k=0}^{M-1} \mathbf{V}'_{d_k}(-x^k) = \mathbf{H}\mathbf{Z}_d^{(\pi)}$  (the subscript  $\pi$  for  $x$  and superscript  $(\pi)$  for  $\mathbf{V}'_{d_k}$  are omitted for simplicity). From this equality, we can deduce  $\mathbf{R}_\pi(x^M \mathbf{I}) = \mathbf{H}\mathbf{X}_\pi(x^M \mathbf{I})$  by substituting  $\mathbf{V}'_{d_k}$  and  $\mathbf{Z}_d^{(\pi)}$  with the corresponding parameters designed in the signing procedure. Since the component  $x^M \mathbf{I}$  is a square matrix with full rank, we know that  $\mathbf{R}_\pi = \mathbf{H}\mathbf{X}_\pi$ . Since  $\mathbf{H}\mathbf{X}_\pi \neq \mathbf{H}\mathbf{X}_j$  for all  $\mathbf{X}_j \in S$ ,  $\mathbf{X}_\pi$  doesn't belong to  $S$  and hence yields a contradiction to the hypothesis that an adversary can produce  $N + 1$  valid signatures. Consequently, our signature scheme  $\mathcal{LRS}$  is linkable.  $\square$

## 5 APQC Based on LRS

In this section, we will introduce how a sender generates the one time address and how a receiver recover the signing key of a transaction. By combining tools introduced here and the linkable ring signature presented in the previous section, we describe the standard transaction of APQC in detail at last.

### 5.1 Stealth Addresses

In CryptoNote, the author suggest using stealth addresses to protect the privacy of the receiver in all transactions. In this protocol, each user is associated with fixed public and private keys. When a sender wants to pay coins to a receiver, a one time address (verifying key which is called the destination address) is generated for the receiver and published as a broadcast by the sender. The receiver then checks every passing transaction with his private key to identify which transaction belongs to him, and then recovers the signing key corresponding to the destination address.

To protect the privacy of receivers, we also design a key-generation protocol to produce stealth addresses which will be used as the verifying and signing key pairs in the linkable ring signature.

### 5.2 Key-Generation Protocol

The key-generation protocol is responsible for three purposes. Firstly, it generates the fixed public and private keys for a user that initially joins the cryptocash system; Secondly, if Alice wants to pay coins to Bob, this protocol produces a new one time destination key for Bob by using random values of Alice and public keys of Bob. Note that the destination key is essentially a verifying key of the linkable ring signature scheme. Thirdly, since Alice broadcasts the transaction labeled with the destination key, the receiver Bob has to efficiently recognize this transaction and recover the corresponding signing key by using the key-generation protocol.

This protocol is formalized as four efficient procedures  $\mathcal{KG}=(\mathbf{Setup}, \mathbf{UKeyGen}, \mathbf{DKeyGen}, \mathbf{DKeyRec})$  which are short forms for user keys generation, destination keys generation, and destination keys recovery respectively.

**Setup**( $1^n, 1^\lambda$ ): On input security parameter, this procedure generates the global parameters  $pp$  for the whole cryptosystem which means this procedure also runs  $\mathcal{LRS.Setup}(1^n)$  and  $\mathcal{ES.Setup}(1^n)$  as subroutines so that the encryption scheme and linkable ring signature scheme are accurately initiated. Let  $(n, m, \mathbf{G}, \mathbf{H}, \mathcal{H}, q, t, N)$  be the system parameters of the linkable ring signature, and  $R = \mathbb{Z}[x]_q / \langle x^n + 1 \rangle$ . Besides that, it chooses a cryptographic hash function  $hash : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$ . Let  $\bar{D} = \{g \in R : \|g\| \leq t/2\}$ .

**UKeyGen**( $pp$ ): When a user wants to join the cryptosystem, he executes this procedure. This procedure first generates the keys for public key encryption scheme  $(epk, esk) \leftarrow \mathcal{ES.KGen}(pp)$ . It then generates a partial key pair of the linkable ring signature scheme  $\mathbf{Y} \leftarrow R^{1 \times m}$ ,  $\mathbf{X} \leftarrow \bar{D}^{m \times m}$ . Note that the norm of the partial signing key  $\mathbf{X}$  is a little smaller than the one of the original linkable ring signature.  $(epk, esk)$  and  $(\mathbf{Y}, \mathbf{X})$  are two pairs of public and private keys which are held by the user.

**DKeyGen**( $pp, epk, \mathbf{Y}$ ): If Alice wants to send coins to Bob who holds keys  $(epk, esk)$ ,  $(\mathbf{Y}, \mathbf{X})$ , she runs the procedure with  $epk$  and  $\mathbf{Y}$ . This procedure samples  $\mathbf{X}_p \leftarrow \bar{D}^{m \times m}$  and generates the destination key  $\mathbf{Y}_d = \mathbf{G}\mathbf{X}_p + \mathbf{Y}$  for Bob.  $\mathbf{X}_p$  is a part of the signing key with respect to the destination key  $\mathbf{Y}_d$ , but no one except Bob can recover the integral signing key. This procedure proceeds to pick an AES secret key  $k$  uniformly at random. It then computes  $c_1 = \mathcal{ES.Enc}_{epk}(k)$  with the public key encryption and computes  $c_2 = \mathbf{AES}_k(hash(epk) \parallel \mathbf{X}_p)$  with the AES algorithm. Finally, it outputs the destination key  $\mathbf{Y}_d$ , and the auxiliary information  $c_1, c_2$ . The brief flow chart for the **DkeyGen** is shown in Fig. 1.

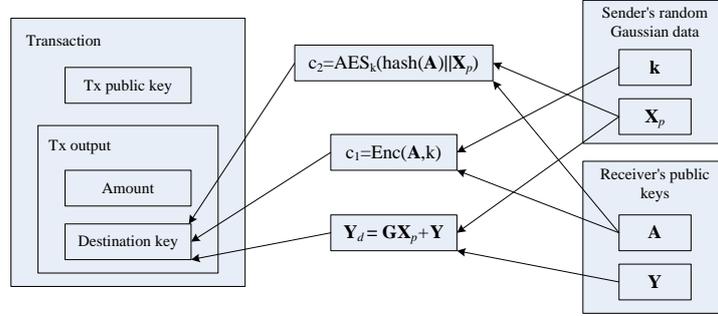


Fig. 1. DKeyGen procedure

**DKeyRec**( $pp, epk, esk, \mathbf{Y}, \mathbf{X}, (\mathbf{Y}_d, c_1, c_2)$ ): Bob runs this procedure to check every passing transaction. If Alice's transaction with Bob as recipient was among them, it will be that (1)  $k = \mathcal{ES.Dec}_{esk}(c_1)$ ; (2)  $(hash(epk) \parallel \mathbf{X}_p) = \mathbf{AES}_k(c_2)$ .

If this procedure finds that the first part of the plaintext of  $c_2$  is not the hash value of Bob's public encryption key  $epk$ , then this procedure aborts and output 0. Otherwise, Bob computes  $\mathbf{X}_d = \mathbf{X}_p + \mathbf{X}$  and  $\mathbf{Y}'_d = \mathbf{G}\mathbf{X}_d$ . If  $\mathbf{Y}'_d = \mathbf{Y}_d$ , this procedure outputs 1 and admits the validity of the destination key  $\mathbf{Y}_d$  and its signing key  $\mathbf{X}_d$ . Since  $\|\mathbf{X}_d\| \leq \|\mathbf{X}_p\| + \|\mathbf{X}\| \leq t$ ,  $\mathbf{X}_d$  is a valid signing key with correspondence to destination key  $\mathbf{Y}_d$ . The process of this procedure is briefly shown in Fig. 2.

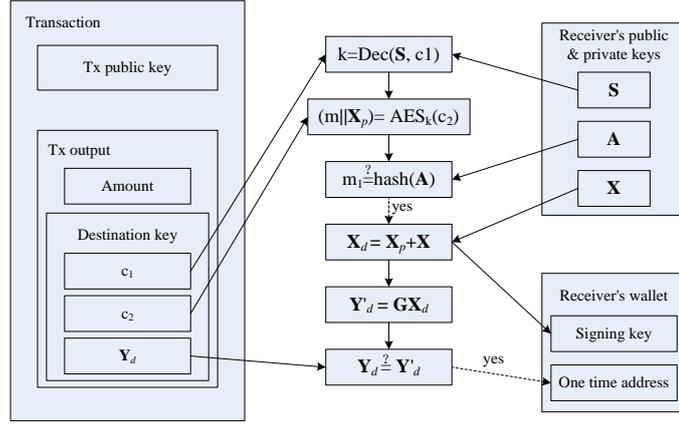


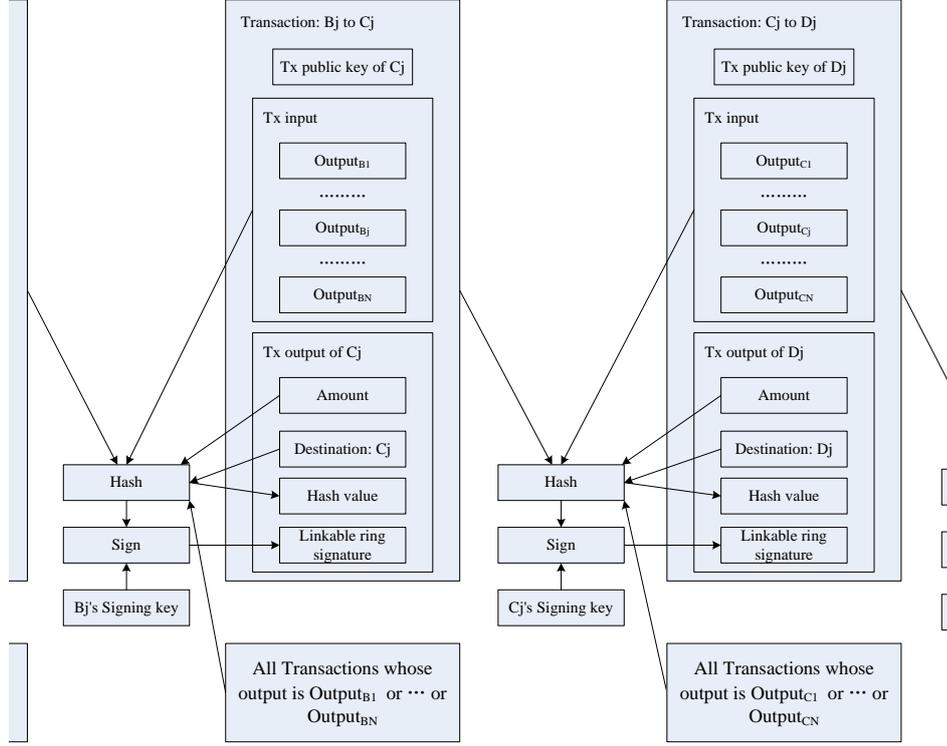
Fig. 2. DKeyRec procedure

### 5.3 Transactions

The standard transaction is briefly shown in Fig. 3.

Bob runs  $\mathcal{KG}.\mathbf{UKeyGen}$  to generate his fixed user key pairs  $(epk_{\text{Bob}}, esk_{\text{Bob}})$ ,  $(\mathbf{Y}_{\text{Bob}}, \mathbf{X}_{\text{Bob}})$  when he initially joins the cryptocash system. Equivalently, Alice holds her key pairs  $(epk_{\text{Alice}}, esk_{\text{Alice}})$ ,  $(\mathbf{Y}_{\text{Alice}}, \mathbf{X}_{\text{Alice}})$ . Alice and Bob also maintain their wallets respectively.

Now, assume that the destination address  $\mathbf{Y}_{B_j}$  and its signing key  $\mathbf{X}_{B_j}$  are in Alice's wallet, and she wants to send coins of this address to Bob. Alice will specify  $N - 1$  foreign outputs  $(\text{Output}_{B_1}, \dots, \text{Output}_{B_{(j-1)}}, \text{Output}_{B_{(j+1)}}, \dots, \text{Output}_{B_N})$  in which the amount is equivalent to that of  $\text{Output}_{B_j}$ . She proceeds to find Bob's public keys  $epk_{\text{Bob}}$  and  $\mathbf{Y}_{\text{Bob}}$  and runs  $\mathcal{KG}.\mathbf{DkeyGen}(pp, epk_{\text{Bob}}, \mathbf{Y}_{\text{Bob}})$  to generate the destination key  $\mathbf{Y}_{C_j}$  and its auxiliary information  $c_1, c_2$  for Bob (see Fig. 1). She then pushes (1) Tx input including  $\{\text{Output}_{B_i}\}_{i=1}^N$  and the amount she sends to Bob, (2) the destination key  $\mathbf{Y}_{C_j}$  and auxiliary information  $c_1, c_2$  she generated for Bob, (3) all previous transactions with output  $\{\text{Output}_{B_i}\}_{i=1}^N$ , into the hash function and signs the hash value  $h$  by running



**Fig. 3.** Transaction chains

$\sigma \leftarrow \mathcal{LRS}.\text{Sign}(pp, \mathbf{X}_{Bj}, h, \mathbf{Y}_{B1}, \dots, \mathbf{Y}_{BN})$ . Finally she broadcasts the transaction which transfer a coin from address  $\mathbf{Y}_{Bj}$  to  $\mathbf{Y}_{Cj}$ .

Bob checks all passing transactions. For each transaction, he extracts the destination key and auxiliary information  $(\mathbf{Y}_d, c_1, c_2)$  and runs the procedure  $\mathcal{KG}.\text{DKeyRec}(pp, epk_{Bob}, \mathbf{Y}_{Bob}, \mathbf{X}_{Bob}, esk_{Bob}, (\mathbf{Y}_d, c_1, c_2))$ . If this transaction is the one that Alice sent to Bob, the forgoing procedure will return the signing key  $\mathbf{X}_{Cj}$  for the destination key is  $\mathbf{Y}_d = \mathbf{Y}_{Cj}$ . If this happens, Bob accepts this transaction and records  $\mathbf{X}_{Cj}, \mathbf{Y}_d$  into his wallet. Bob can later spend the coin bound with destination address  $\mathbf{Y}_d$  because he has the secret signing key  $\mathbf{X}_{Cj}$ .

## 6 Conclusions and Future Works

While a lot of lattice-based ring signature and standard signature have recently been designed, linkable ring signature over lattices has not been to the best of our knowledge. Since there are many similarities between the DLP and the ISIS problem, we can easily transform a DLP based linkable ring signature into a lattices based one. However, such kind of linkable ring signature is too cumbersome

to be sued for a cryptocash protocol. In this paper, we construct a linkable ring signature from lattices where the size of a signature is  $O(\log N)$ , where  $N$  is the number of participants of the ring. Based on the proposed logarithmic size linkable ring signature scheme, an anonymous post-quantum cryptocash system is also presented. In order to generate stealth addresses (verifying keys) and recover corresponding signing keys for the linkable ring signature, we also propose a key-generation protocol as a subroutine of the cryptocash protocol. By combining all those techniques together, our cryptocash protocol obtains a new level anonymity comparing to the original Bitcoin protocol. Furthermore, the new designed cryptocash protocol has the potential of resist attacks from quantum.

Next, we shall implement a full cryptocash system to test the communication and computation costs. And if possible, we would like to contribute our system to the cryptocash community for public usage.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61379154 and 61672550).

## References

1. Aharonov, D., Regev, O.: Lattice problems in  $NP \cap coNP$ . *J. ACM* 52(5), 749–765 (Sep 2005), <http://doi.acm.org/10.1145/1089023.1089025>
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: *ACM Symposium on Theory of Computing – STOC 1996*. pp. 99–108. ACM (1996), <http://doi.acm.org/10.1145/237814.237838>
3. Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to better — how to make bitcoin a better currency. In: *Financial Cryptography and Data Security – FC 2012*. pp. 399–414. Springer Berlin Heidelberg (2012), [https://doi.org/10.1007/978-3-642-32946-3\\_29](https://doi.org/10.1007/978-3-642-32946-3_29)
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: *Advances in Cryptology — ASIACRYPT 2001*. pp. 566–582. Springer Berlin Heidelberg (2001), [https://doi.org/10.1007/3-540-45682-1\\_33](https://doi.org/10.1007/3-540-45682-1_33)
5. Brakerski, Z., Kalai, Y.T.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. *Cryptology ePrint Archive, Report 2010/086* (2010), <http://eprint.iacr.org/2010/086>
6. Cai, J.Y., Nerurkar, A.P.: An improved worst-case to average-case connection for lattice problems. In: *Symposium on Foundations of Computer Science – FOCS 1997*. pp. 468–477. IEEE (Oct 1997), <http://doi.org/10.1109/SFCS.1997.646135>
7. Cayrel, P.L., Lindner, R., Rückert, M., Silva, R.: A lattice-based threshold ring signature scheme. In: *Progress in Cryptology – LATINCRYPT 2010*. pp. 255–272. Springer Berlin Heidelberg (2010), [https://doi.org/10.1007/978-3-642-14712-8\\_16](https://doi.org/10.1007/978-3-642-14712-8_16)
8. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: *Advances in Cryptology – CRYPTO 2013*. pp. 40–56. Springer Berlin Heidelberg (2013), [https://doi.org/10.1007/978-3-642-40041-4\\_3](https://doi.org/10.1007/978-3-642-40041-4_3)
9. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: *Public Key Cryptography – PKC 2007*. pp. 181–200. Springer Berlin Heidelberg (2007), [https://doi.org/10.1007/978-3-540-71677-8\\_13](https://doi.org/10.1007/978-3-540-71677-8_13)

10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: ACM Symposium on Theory of Computing – STOC 2008. pp. 197–206. ACM (2008), <http://doi.acm.org/10.1145/1374376.1374407>
11. Goldreich, O., Goldwasser, S.: On the limits of non-approximability of lattice problems. In: ACM Symposium on Theory of Computing – STOC 1998. pp. 1–9. ACM (1998), <http://doi.acm.org/10.1145/276698.276704>
12. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Advances in Cryptology - EUROCRYPT 2015. pp. 253–280. Springer Berlin Heidelberg (2015), [https://doi.org/10.1007/978-3-662-46803-6\\_9](https://doi.org/10.1007/978-3-662-46803-6_9)
13. Halevi, S.: A sufficient condition for key-privacy. Cryptology ePrint Archive, Report 2005/5 (2005), <http://eprint.iacr.org>
14. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In: Advances in Cryptology – EUROCRYPT 2016. pp. 1–31. Springer Berlin Heidelberg (2016), [https://doi.org/10.1007/978-3-662-49896-5\\_1](https://doi.org/10.1007/978-3-662-49896-5_1)
15. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Information Security and Privacy – ACISP 2004. pp. 325–335. Springer Berlin Heidelberg (2004), [https://doi.org/10.1007/978-3-540-27800-9\\_28](https://doi.org/10.1007/978-3-540-27800-9_28)
16. Liu, J.K., Wong, D.S.: Linkable ring signatures: Security models and new schemes. In: Computational Science and Its Applications – ICCSA 2005. pp. 614–623. Springer Berlin Heidelberg (2005), [https://doi.org/10.1007/11424826\\_65](https://doi.org/10.1007/11424826_65)
17. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Advances in Cryptology – EUROCRYPT 2012. pp. 738–755. Springer Berlin Heidelberg (2012), [https://doi.org/10.1007/978-3-642-29011-4\\_43](https://doi.org/10.1007/978-3-642-29011-4_43)
18. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Automata, Languages and Programming – ICALP 2006. pp. 144–155. Springer Berlin Heidelberg (2006), [https://doi.org/10.1007/11787006\\_13](https://doi.org/10.1007/11787006_13)
19. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: Symposium on Foundations of Computer Science – FOCS 2002. pp. 356–365 (2002), [doi.org/10.1109/SFCS.2002.1181960](https://doi.org/10.1109/SFCS.2002.1181960)
20. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *computational complexity* 16(4), 365–411 (Dec 2007), <https://doi.org/10.1007/s00037-007-0234-9>
21. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing* 37(1), 267–302 (2007), <https://doi.org/10.1137/S0097539705447360>
22. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: Symposium on Security and Privacy – SP 2013. pp. 397–411 (May 2013), [doi.org/10.1109/SP.2013.34](https://doi.org/10.1109/SP.2013.34)
23. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. URL: <http://www.bitcoin.org/bitcoin.pdf> (2012)
24. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. *Future Internet* 5(2), 237–250 (2013), <http://doi.org/10.3390/fi5020237>
25. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Security and Privacy in Social Networks. pp. 197–223. Springer New York (2013), [https://doi.org/10.1007/978-1-4614-4139-7\\_10](https://doi.org/10.1007/978-1-4614-4139-7_10)
26. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Advances in Cryptology — ASIACRYPT 2001. pp. 552–565. Springer Berlin Heidelberg (2001), [https://doi.org/10.1007/3-540-45682-1\\_32](https://doi.org/10.1007/3-540-45682-1_32)

27. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Financial Cryptography and Data Security – FC 2013. pp. 6–24. Springer Berlin Heidelberg (2013), [https://doi.org/10.1007/978-3-642-39884-1\\_2](https://doi.org/10.1007/978-3-642-39884-1_2)
28. Saberhagen, N.v.: Cryptonote v 2. 0. HYPERLINK <https://cryptonote.org/whitepaper.pdf> (2013)
29. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Advances in Cryptology – ASIACRYPT 2009. pp. 617–635. Springer Berlin Heidelberg (2009), [https://doi.org/10.1007/978-3-642-10366-7\\_36](https://doi.org/10.1007/978-3-642-10366-7_36)
30. Wang, C., Wang, H.: A new ring signature scheme from ntru lattice. In: Computational and Information Sciences – ICCIS 2012. pp. 353–356. IEEE (Aug 2012), [doi.org/10.1109/ICCIS.2012.31](https://doi.org/10.1109/ICCIS.2012.31)
31. Wang, J., Sun, B.: Ring signature schemes from lattice basis delegation. In: Information and Communications Security – ICICS 2011. pp. 15–28. Springer Berlin Heidelberg (2011), [https://doi.org/10.1007/978-3-642-25243-3\\_2](https://doi.org/10.1007/978-3-642-25243-3_2)

## Appendix A. Short Linkable Ring Signature Based on ECDLP

For an integer  $i$ ,  $i_j$  denotes its  $j^{\text{th}}$  bit.  $\delta_{i\ell}$  is Kronecker’s delta. Let  $N$  be the size of ring and  $n = \log N$ . Define  $f_{j,1} = f_j = \ell_j e + a_j = \delta_{1\ell_j} e + a_j$ , and  $f_{j,0} = e - f_j = (1 - \ell_j)e - a_j = \delta_{0\ell_j} e - a_j$ . For each  $i$  the product  $\prod_{j=1}^n f_{j,i_j}$  is a polynomial in the indeterminate  $e$  of the form

$$p_i(e) = \prod_{j=1}^n (\delta_{i_j \ell_j} e) + \sum_{k=0}^{n-1} p_{i,k} e^k = \delta_{i\ell} e^n + \sum_{k=1}^{n-1} p_{i,k} e^k.$$

Here,  $p_{i,k}$  is the coefficient of the  $k^{\text{th}}$  degree term of the polynomial  $p_i(e)$ , and can be efficiently computed when  $\{a_j\}_{j=1}^n$  and  $\ell$  are given.

The linkable ring signature based on ECDLP consists of five efficient procedures (**Setup**, **KGen**, **Sign**, **Vry**, **Link**).

**Setup**( $1^\lambda$ ): Let  $E$  be an elliptic curve defined over a finite field  $\mathbb{F}_q$ . Let  $G \in E$  be a point of prime order  $p$ , here  $|p| = \lambda$  and let  $\mathbb{G}$  be the prime order subgroup of  $E$  generated by  $G$ . Choose another element  $H \in \mathbb{G}$  randomly. Let  $\mathcal{H} : \mathbb{G} \rightarrow \mathbb{Z}_p$  be a cryptographic hash function. The output of this procedure is  $pp = (\mathbb{G}, G, H, p, q, \mathcal{H})$ .

**KGen**( $pp$ ): For the  $i^{\text{th}}$  user, this procedure chooses the signing key  $x_i \in \mathbb{Z}_p$  uniformly at random and computes the verifying key  $Y_i = x_i G$ . It outputs  $(x_i, Y_i)$  as the key pair of the  $i^{\text{th}}$  user.

**Sign**( $pp, x_\ell, \mu, L$ ): Let  $L = (Y_0, Y_1, \dots, Y_{N-1})$  be the ensemble of the ring. On input the message  $\mu$ , the  $\ell^{\text{th}}$  user’s signature on behalf of  $L$  is generated as follows

- Compute  $I_\ell = x_\ell H$ .
- For  $j$  from 1 to  $n$ ,
  - choose  $r_j, a_j, s_j, t_j, \rho_k \leftarrow \mathbb{Z}_p$  at random.
  - compute  $C_{\ell_j} = \ell_j H + r_j G$ ,

- compute  $C_{a_j} = a_j H + s_j G$ ,
  - compute  $C_{b_j} = a_j \ell_j H + t_j G$ ,
  - compute  $C_{d_k} = (\sum_{i=0}^{N-1} p_{i,k} Y_i) + \rho_k G$ , where  $k = j - 1$ ,
  - compute  $C'_{d_k} = \rho_k H$ , for  $k = j - 1$ .
- Let  $\mathbf{a} = \{C_{\ell_j}, C_{a_j}, C_{b_j}, C_{d_{j-1}}, C'_{d_{j-1}}\}_{j=1}^n$  and compute  $e = \mathcal{H}(pp, u, L, \mathbf{a}, I_\ell)$
- For  $j$  from 1 to  $n$ , compute
- $f_j = e \ell_j + a_j$ ,
  - $z_{a_j} = e r_j + s_j$ ,
  - $z_{b_j} = (e - f_j) r_j + t_j$ ,
  - $z_d = e^n x_\ell - \sum_{k=0}^{n-1} e^k \rho_k$ .
- Let  $\mathbf{b} = \{f_j, z_{a_j}, z_{b_j}\}_{j=1}^n$ . Publish  $\sigma = \{\mathbf{a}, \mathbf{b}, z_d, I_\ell, L\}$  as the signature of the  $\ell^{th}$  user.

**Vry**( $pp, \mu, \sigma, L$ ):

- Compute  $e = \mathcal{H}(pp, \mu, L, \mathbf{a}, I_\ell)$
- For  $j$  from 1 to  $n$ , consider the following equalities
- $e C_{\ell_j} + C_{a_j} = f_j H + z_{a_j} G$ ,
  - $(e - f_j) C_{\ell_j} + C_{b_j} = z_{b_j} G$ ,
- If any one of them doesn't hold, output 0 and abort.
- If the equality  $e^n I_\ell + \sum_{k=0}^{n-1} (-e^k) C'_{d_k} = z_d H$  doesn't hold, output 0 and abort.
- Inspect whether  $\sum_{i=0}^{N-1} (\prod_{j=1}^n f_{j,i_j}) Y_i + \sum_{k=0}^{n-1} (-e^k) C_{d_k} = z_d G$ . If it is not, output 0 and abort; otherwise output 1.

**Link**( $pp, \sigma, \sigma'$ ): For any two signatures  $\sigma_1 = (\dots, I_1, L_1)$  and  $\sigma_2 = (\dots, I_2, L_2)$ , if  $I_1 = I_2$ , return 1 (linked) for concluding that they are generated by the same signer; otherwise, return 0 (unlinked). Since  $H$  is the global parameter and  $I_\ell = x_\ell H$  is an ingredient to verify the valid signature,  $x_i$  can only sign one message during the whole system life.

Using the above logarithmic size linkable ring signature to replace the linkable ring signature scheme in Monero, we can get a more efficient Monero system.